

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-315954

(43)Date of publication of application : 14.11.2000

(51)Int.Cl.

H03M 7/30

(21)Application number : 2000-078069

(71)Applicant : LUCENT TECHNOL INC

(22)Date of filing : 21.03.2000

(72)Inventor : BENTLEY JON LOUIS
MCILROY MALCOLM DOUGLAS

(30)Priority

Priority number : 99 273840

Priority date : 22.03.1999

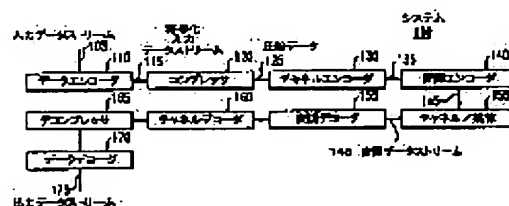
Priority country : US

(54) METHOD FOR COMPRESSING INPUTTED DATA STREAM AND DEVICE THEREFOR

(57)Abstract:

PROBLEM TO BE SOLVED: To improve data compression to a larger input data stream by crossing the input data stream and comparing an intermediate fingerprint calculated from a specified character set of the input data stream with a previously calculated and stored fingerprint.

SOLUTION: The input data stream 105 is supplied to an input data encoder 110, the input data stream is preprocessed and encoded. The encoded input data stream 115 is compressed to compressed data 125 by a compressor 120. A data stream from a channel decoder 160 is decompressed by a decompressor 165, decoded by a data decoder 170 and an output data stream 175 is created. And the input data stream is divided into groups of blocks, after that, the fingerprints are calculated for respective blocks and stored. Data compression structure to recognize correlation between different strings is further used.



LEGAL STATUS

[Date of request for examination]

23.04.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P) (12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-315954
(P2000-315954A)
(43) 公開日 平成12年11月14日 (2000.11.14)

(51) IntCl.⁷
H 03 M 7/30 識別記号
F I
H 03 M 7/30 Z

審査請求 未請求 請求項の数33 OL (全 17 頁)

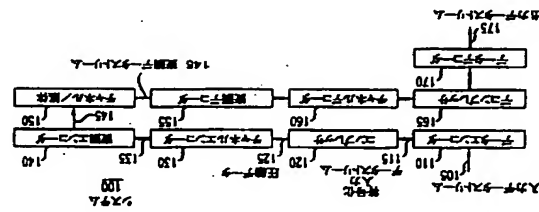
(21) 出願番号 特開2000-7806X (P2000-7806X)
(22) 出願日 平成12年9月21日 (2000.9.21)
(31) 発明者氏名 09/273840
(32) 優先日 平成11年9月22日 (1999.9.22)
(33) 優先地 米国 (U S)
(71) 出願人 598077259
ルーセント テクノロジーズ インコーポ
レイテッド
Lucent Technologies
Inc.
アメリカ合衆国 07974 ニュージャージー
一、マレービル、マウンテン アベニュー
600-700
(74) 代理人 100081053
井理士 三俣 弘文

最終頁に続く

(54) 【発明の名称】 入力データストリームの圧縮方法とその装置

(57) 【要約】

【課題】 記憶装置の条件や伝送時間を減らすために更に大きく圧縮する方法を提供する。
【解決手段】 圧縮プロセス (例えば、いずれかのLempel-Ziv圧縮方式) を適用する前に入力データの初期評価として入力データストリームのより長い履歴およびより長いコンテキストリングを用いる。すなわち、通常の圧縮プロセスが所望の圧縮を行うために比較的短い (最も最近の数千バイト) 入力データの履歴を用いるが、より長いコンテキストリングシーケンスを用いることにより長い履歴を用いることが組み合わさって、圧縮効率を増加させる。これは特に、繰り返し発生する文字を多く有する長い入力データストリームを圧縮する際に有効である。本発明に従うと、抽出した長コンテキストリングに対してストリーミングマッピングを適用することにより入力データをブリアプロセス (前処理) する。



【特許請求の範囲】

【請求項1】 入力データストリームを圧縮する方法であって、

(A) 入力データストリームを複数のデータブロックへと分割するステップと、
(B) 複数のフィンガプリントを計算するステップと、ここで、各フィンガプリントは、前記複数のデータブロックの異なる1つのデータブロックに対応し、
(C) 前記複数のフィンガプリントの各フィンガプリントは、前記複数のデータブロックの異なる1つのデータブロックに対応し、入力データストリームを前記複数のフィンガプリントと比較するステップと、
(D) 特定のフィンガプリントと入力データストリームの間にマッチが出現すれば、そのマッチを入力データストリーム内のそのマッチを符号化するステップと、
(E) 符号化したデータストリームを圧縮データストリームへと圧縮するステップとを有することを特徴とする方法。
【請求項2】 前記入力データストリームは一連の文字からなることを特徴とする請求項1記載の方法。
【請求項3】 前記ステップ (C) は、一連の文字の個々の文字の関数として入力データストリームを順に検出し、その個々の文字のそれぞれの関数として中間フィンガプリントを計算することを特徴とする請求項2記載の方法。

【請求項4】 (F) 前記複数のフィンガプリントと共に前記複数のデータブロックをデータ構造に記憶するステップとを有することを特徴とする請求項2記載の方法。

【請求項5】 前記ステップ (C) は、特定のフィンガプリントにマッチする一連の文字から最も長いマッチしたシーケンスを識別することを特徴とする請求項3記載の方法。

【請求項6】 前記ステップ (C) は、中間フィンガプリントを特定のフィンガプリントと比較することを特徴とする請求項3記載の方法。

【請求項7】 前記ステップ (D) は、最も長いマッチしたシーケンスの入力データストリームにおける開始位置および最も長いマッチしたシーケンスの長さを符号化することを特徴とする請求項5記載の方法。

【請求項8】 複数の文字を含む入力データストリームを圧縮コードストリームへと処理する方法であって、
(A) 入力データストリームを複数のブロックに分割するステップと、ここで、各ブロックは、複数の文字のうち特定の数の文字を含み、
(B) 複数のフィンガプリントを計算するステップと、ここで、各フィンガプリントは、前記複数のデータブロックの異なる1つのデータブロックに対応し、
(C) 各フィンガプリントは複数のブロックの異なる1つのブロックに対応し、特定のフィンガプリントと特

定のブロックのいずれかの部分との間にマッチが出現するかを判断するために、複数のブロックを複数のフィンガプリントと比較するステップと、
(D) 出現したマッチそれぞれに対して入力データストリームにて識別子を符号化するステップと、ここで、この識別子は、特定のブロックのマッチ部分の入力データストリームにおける開始位置およびマッチ部分の長さを含み、
(E) 符号化された入力データストリームを圧縮コードストリームへと圧縮するステップとを有することを特徴とする方法。
【請求項9】 符号化された入力データストリームの圧縮は、Lempel-Zivコーディング技術に従って行われることを特徴とする請求項1、8記載の方法。
【請求項10】 前記ステップ (C) は、
(a) ウィンドウサイズを選択するステップと、
(b) ウィンドウサイズの関数として複数のブロックを検出するステップとを有することを特徴とする請求項8記載の方法。
【請求項11】 ウィンドウサイズは、特定の数の文字であることを特徴とする請求項10記載の方法。
【請求項12】 前記ステップ (C) は、
(c) ブロックに含まれる特定の数の文字それぞれにおけるブロックに対する中間フィンガプリントを計算するステップと、
(d) 中間フィンガプリントを特定のフィンガプリントと比較するステップとを有することを特徴とする請求項11記載の方法。
【請求項13】 前記複数のブロックの各ブロックはブロックサイズが等しいことを特徴とする請求項11記載の方法。
【請求項14】 前記ブロックサイズは10~1000文字の範囲であることを特徴とする請求項13記載の方法。
【請求項15】 前記検出 (b) は、ブロックに含まれる特定の数の文字を順に検出することを特徴とする請求項12記載の方法。
【請求項16】 (F) 前記複数のフィンガプリントと共に前記複数のデータブロックをデータ構造に記憶するステップを有し、このデータ構造は、各フィンガプリントをその異なる1つの対応するブロックと共に記憶することを特徴とする請求項8記載の方法。
【請求項17】 前記開始位置は、入力データストリームにおける文字の位置であり、マッチした部分の長さは文字の数であることを特徴とする請求項16記載の方法。
【請求項18】 (A) 入力データストリームを受信し、複数のブロックへと分割する受信器と、ここで、各ブロックは、入力データストリームからの複数の文字の特定の数の文字を含み、

(B) 複数のフィンガプリントを計算するエンコーダと、ここで、

各フィンガプリントは、前記複数のブロックの異なる1つのブロックに対応し、このエンコーダは、複数のブロックを複数のフィンガプリントと比較し、特定のフィンガプリントと特定のブロックのいずれかの部分の間でマッチが出現するかを判断し、出現した各マッチに対してそのマッチの入力データストリームにおける識別子を符号化し、

(C) 符号化された入力データストリームを圧縮コードストリームへと圧縮するコンプレッサとを有することを特徴とするデータ圧縮を行う装置、

【請求項19】 (D) 複数のブロックを複数のフィンガプリントと共にデータ構造に記憶するメモリを更に有し、

このデータ構造は、各フィンガプリントをその異なる1つの対応するブロックと共に記憶することを特徴とする請求項18記載のデータ圧縮を行う装置、

【請求項20】 前記複数のブロックの比較は、(a) ブロックに含まれる特定の数の各文字におけるブロックに対して中間フィンガプリントを計算し、

(b) 中間フィンガプリントを特定のフィンガプリントと比較することを特徴とする請求項18記載のデータ圧縮を行う装置、

【請求項21】 前記識別子は、特定のブロックのマッチした部分の入力データストリームにおける開始位置およびマッチした部分の長さを含むことを特徴とする請求項19記載のデータ圧縮を行う装置、

【請求項22】 前記複数のブロックの各ブロックは、ブロックサイズが小さいことを特徴とする請求項21記載のデータ圧縮を行う装置、

【請求項23】 前記開始位置は、入力データストリームにおける文字の位置であり、前記マッチした部分の長さは、文字の数であることを特徴とする請求項22記載のデータ圧縮を行う装置、

【請求項24】 前記コンプレッサ (C) は、圧縮コードストリームへと符号化された入力データストリームを圧縮するのにLempel-Zivコーディングを用いることを特徴とする請求項18記載のデータ圧縮を行う装置、

【請求項25】 デジタルデータストリームを圧縮データ形式で記憶するデータ記憶システムであって、

(A) デジタルデータストリームを複数のデータブロックへと分割する手段と、ここで、各ブロックは複数の文字を有し、

(B) 複数のフィンガプリントを計算する手段と、ここで、各フィンガプリントは、前記複数のデータブロックの異なる1つのデータブロックに対応し、

(C) 複数のブロックを識別する手段と、

(D) ブロックの複数の各文字における各ブロックに対して中間フィンガプリントを計算する手段と、

(E) 中間フィンガプリントを複数のフィンガプリントの特定のフィンガプリントと比較する手段と、

(F) その特定のフィンガプリントと中間フィンガプリントの間でマッチが出現すれば、そのマッチをデジタルデータストリームにおいて符号化する手段と

(G) 符号化されたデータストリームを圧縮データ形式に圧縮する手段と、を有することを特徴とするデータ記憶システム、

【請求項26】 (H) 複数のデータブロックを複数のフィンガプリントと共にデータ構造に記憶する手段を更に有することを特徴とする請求項25記載のデータ記憶システム、

【請求項27】 前記識別する手段 (C) は、各ブロックの複数の各文字の関数としてブロックを順に識別することを特徴とする請求項26記載のデータ記憶システム、

【請求項28】 前記圧縮する手段 (G) は、符号化されたデジタルデータストリームを圧縮データ形式に圧縮するためにLempel-Zivコーディングを用いることを特徴とする請求項27記載のデータ記憶システム、

【請求項29】 前記符号化されたマッチは、特定の部分のマッチした部分のデジタルデータストリームにおける開始位置およびマッチした部分の長さを有することを特徴とする請求項25記載のデータ記憶システム、

【請求項30】 圧縮デジタル番号を処理する装置であって、

(a) 入力デジタルデータストリームを複数のデータブロックへと分割し、

(b) それぞれが複数のデータブロックの異なる1つのデータブロックに対応する複数のフィンガプリントを計算し、

(c) 入力デジタルデータストリームを複数のフィンガプリントと比較し、

(d) 特定のフィンガプリントと入力デジタルデータストリームの間でマッチが出現すれば、入力デジタルデータストリームにおいてそのマッチを符号化し、

(e) 符号化された入力デジタルデータストリームを圧縮デジタル番号に圧縮し、

(f) 圧縮デジタル番号を通信チャネルへ、供給することにより作られ、

(A) 通信チャネルから圧縮デジタル番号を受信する受信器と、

(B) 受信した圧縮デジタル番号を脱圧縮し、その脱圧縮したデジタル番号から入力デジタルデータストリームを回復するデコンプレッサとを有することを特徴とする圧縮デジタル番号を処理する装置、

【請求項31】 前記複数のブロックの各ブロックは、複数の文字を含み、前記入力デジタルデータストリームの比較 (c) は、

(g) ブロックの複数の各文字におけるブロック

に対する中間フィンガプリントを計算し、

(h) 中間フィンガプリントを特定のフィンガプリントと比較することを特徴とする請求項30記載の装置、

【請求項32】 前記複数のブロックの各ブロックは、ブロックサイズが小さいことを特徴とする請求項31記載の装置、

【請求項33】 前記複数のフィンガプリントの各フィンガプリントは対応するデータブロックの特定の文字群を有することを特徴とする請求項31記載の装置、

【請求項34】 以下の命令を含む複数の命令が記憶された記憶装置が読み取り可能な媒体、すなわち、

(A) 媒体により実行されると、入力データストリームを圧縮コードストリームへと処理させ、

(B) 入力データストリームを複数のブロックへと分割させ、ここで、各ブロックは複数の文字の特定の数の文字を含み、

(C) 複数のフィンガプリントを計算させ、ここで、各フィンガプリントは、前記複数のブロックの異なる1つのブロックに対応し、

(D) 複数のブロックを複数のフィンガプリントと比較させ、

(E) 特定のフィンガプリントと特定のブロックのいずれかの部分の間でマッチが出現するかどうかを判断させ、

(F) 出現した各マッチに対して、入力データストリームにおけるそのマッチの識別子を符号化させ、ここで、この識別子は、特定のブロックのマッチした部分の入力データストリームにおける開始位置およびマッチした部分の長さを有し、

(G) 符号化された入力データストリームを圧縮コードストリームへと圧縮させる命令、

【請求項35】 前記複数のフィンガプリントの各フィンガプリントは、その対応するブロックの特定の文字群を有することを特徴とする請求項34記載の装置が読み取り可能な媒体、

【請求項36】 前記比較 (D) は、ブロックにおける文字の関数として行われ、ブロックに対する中間フィンガプリントは、前記複数の文字の各文字にて計算され、

前記特定のフィンガプリントと比較されることを特徴とする請求項34記載の装置が読み取り可能な媒体、

【請求項37】 前記圧縮は、Lempel-Zivコーディング技術に従って行われることを特徴とする請求項36記載の装置が読み取り可能な媒体、

【請求項38】 (A) それぞれが複数の文字からなる複数のブロックへと入力データストリームを分割し、

(B) それぞれが前記複数のブロックの異なる1つのブロックに対応する複数のフィンガプリントを計算し、

(C) ブロックの複数の各文字の関数として入力ストリームを識別し、

フィンガプリントを計算し、

(E) 中間フィンガプリントを複数のフィンガプリントと比較し、

(F) 複数のフィンガプリントの特定のフィンガプリントと中間フィンガプリントの間でマッチが出現するかを判断し、

(G) 出現した各マッチに対して、ブロックのマッチした部分の入力データストリームの開始位置およびマッチした部分の長さを有し、入力データストリーム内の識別子を符号化し、

(H) 符号化された入力データストリームを圧縮コードストリームへと圧縮し、

(I) 圧縮コードストリームを符号化記憶媒体に供給し、

(J) 記憶符号を記憶媒体上に記録する動作からなるプロセスを用いて作成された記憶媒体、

【発明の詳細な説明】
[0001]

【発明の属する技術分野】 本発明は、データ記憶および通信システムに関し、特に、それらシステムの容量および利用を改善することに関する。

【従来の技術】 従来のデータ圧縮技術およびシステムはデジタルデータストリームを圧縮コードストリームへと符号化し、圧縮コードストリームを対応する元のデータストリームへとデコードして戻す。ここでコードストリームを「圧縮」としているが、それは、コードストリームが通常、元のデータストリームに含まれる符号の数よりも少ない数のコードであるからである。このような小さいコードは元のデータよりも少ない量のメモリに記憶することができる。

【0003】 また、圧縮コードストリームは、圧縮されていない元のデータよりも短い時間で通信システム (例えば、有線、無線、光ファイバ通信システム) にて送信することができ、今日の通信ネットワークにおいては、コンテンツ交換の量が相対的に増大しており、データ送信および記憶容量の必要性が今までかつてないほど増えている。このように、データ圧縮は現代の通信プロトコルおよび通信ネットワークにおいて重要な役割を担っている。

【0004】 データ圧縮に有用な2つのクラスの圧縮技術は、周知のように、いわゆる特殊用途圧縮と汎用圧縮である。特殊用途圧縮技術は特殊な種類のデータを圧縮するために設計され、実装するのに比較的コストであることが多い。例えば、周知の特殊用途圧縮技術として、ランレングス符号化、ゼロ圧縮、圧縮符号化、パターンの圧縮符号化、パターンの置き換え技術がある。

【0005】 これら技術により、一般に、比較的小さい圧縮比となる。なぜなら、一般的特性および冗長性を有するデータを通常圧縮するからである。圧縮比とは、

データストリーム（本発明に従って符号化された元の入カデータストリーム）が周知の圧縮アルゴリズムのいずれか（例えば、Lempel-Ziv圧縮）を圧縮される（290）。

（0041）以下の導コードは、上述のような本発明の原理に従うフィンガプリントの比較および符号化を記したものである。変数inは、フィンガプリントを表し、関数checkformalchはハッシュテーブルにおけるフィンガプリントをルックアップし、マッチを見つけたらそのマッチを符号化する。

```
コード1
initialize fp
for (ib; (in; i++))
  if (!xib == 0)
    store (fp, i)
  a[i]を含まa[i-b]を含まないようiにipを更新
  checkformalch(fp, i)
```

（0042）上の導コードは、本発明を装飾するプロセスにおける実行をするために、多くのプログラム（例えば、C書庫プログラム）にて開発するのに用いることができる。例えば、図9～10は、本発明に従ってデータ圧縮するC書庫ソースプログラム900を示す。ソースプログラム900は、このソースコードプログラム900全体で用いる特定のデータ型、変数、データ構造を規定するプログラム命令を含むプログラムソースコード部分910を含む。

（0043）プログラムソースコード部分920は、上述のように本発明に従ってストリングマッピング動作を実行するプログラム命令を含む。プログラムソースコード部分930は、上述のように本発明に従って計算されるフィンガプリントを保持するのに用いるハッシュテーブルデータ構造を定義するのを支援するプログラム命令である。プログラムソースコード部分940、950（図10）は、上述のように本発明に従って符号化データファイルを作るために圧縮を完成させるプログラム命令を含む。

（0044）C書庫ソースプログラム900は、生来的に例示的であり本発明を理解するのを支援するために提供してある。本発明を具現化する他のプログラムを本発明の範囲から外れずに開発することは当業者であればである。

（0045）上述の本発明の多くの原理を更に説明するため、図4は、一連の短い非圧縮入カデータストリーム400を、本発明に従って符号化された対応する一連の符号化入カデータストリーム410と共に示してある。図4を調べることににより、入カデータストリーム420～450はそれぞれ、符号化出力460～490それぞれに示すようにマッピングストリングの符号化された級現によって処理されている。

（0046）また、図5、6には本発明を更に説明する

ために、大きな入カデータファイル、具体的には、アメリカ合衆国の憲法に対する本発明の原理の適用を示している。図5は、憲法の選択部分からなる入カデータファイル500を示している。本発明に従う憲法のテキストの圧縮は、同じ長いストリングが頻りに現れることを考へると特に本発明の利点を顕著に現れる。例えば、入カデータファイル500のテキスト部分510、520、530、540はほぼ同じ長い繰り返しストリングを含む。

（0047）従って、上述の本発明の多くの原理を適用すると、図6に示した符号化データファイル600を得る。600は符号化部分610、620、630を含む。これらそれぞれは入カデータファイル500のテキスト部分510、520、530に対応する。符号化部分610、620、630はそれぞれ、エンコーディング（例えば、エンコーディング635～690）を含む。これらは、本発明に従って導かれ、ブロックサイズ＝20を用い、更に圧縮比を増しデータ伝送レートをより効率的にするために更に圧縮されることができ、

（0048）例えば、エンコーディング635は、文字位置391に始まる47文字のマッピングシグネチャ（すなわち、マッピングストリング）が抽出され符号化されたことを示す。同様に、例えば、エンコーディング675は、文字位置2439に始まる103文字のマッピングシグネチャが抽出され符号化されたことを示す。（0049）圧縮プロセスを適用する前に入カデータの初期評価（プリプロセス）においてフィンガプリントを用いることを認識することにより、入カデータストリームの長い問題および長いコメントストリングを用いることに基づいて比較的低い圧縮比を実現することができ、重要なことに、本発明は配電装置の必要条件を余り増加させず、いずれの特定の圧縮技術にも依存しない。すなわち、本発明は、相当な圧縮比を実現する多様な周知の圧縮アルゴリズムを用い、配電装置の必要条件や伝送時間を減らすことができる。

（0050）圧縮テスト結果を記す前に、本発明を用いて得られる符号化データファイルの圧縮率を議論する。図7は、本発明の更なる原理に従ってデータを圧縮する動作700の流れ図である。本発明による符号化データファイル（例えば、符号化データファイル600）から、この図では文字毎のベースで図々の文字「c」を取り出す（710）。

（0051）特定の文字cがシンボル「<」とマッチするかどうかの判断を行う（720）。マッチしなければ、脱圧縮プロセスに従って出力ファイルへと文字cが書き込まれる（730）。符号化データファイルから次の文字が取り出される。マッチすれば、符号化データファイルから次の文字が取り出される（740）。再び、特定の文字cがシンボル「<」とマッチするかどうかの判断をする（750）。マッチすれば、出力ファイルにシンボル「<」が書き込まれる（760）。次の文字が取り

本発明の原理と組み合わせる周知の「gzip」圧縮アルゴリズム（周知のGNUによるLZW77の装束であるgzip）を適用した結果を示す。「comlgzip.k」（850）は、列820のサイズに対する列840の割合を示す。

例えば、列820のサイズを適用し、元のファイルサイズを7%の割合で減らすことができた。図よりブロックサイズが減ると、最適でない点（効率的な圧縮にはブロックサイズが小さすぎ）に到達するまで圧縮の割合は増えている。また、「total.k」の見出しを有する列860は、元のファイルと比較した列840の割合を示し、bの最適な選択は、comlgzipのファイルサイズとの減少量を超えて22.5%のファイルサイズの減少量となっているところの31kbyteである。

（0058）本明細書における説明は単に本発明の原理の例を示したのみである。当業者は、本発明の範囲を外れずに多くの構成を考へることができ得るであろう。特に、多くの構成を考へることができ得るであろう。特許請求範囲の記載において特定の機能を実行する手段として表現したいずれの要素も、その機能を実行するいずれの機能をも表すように意図してある。例えば、（a）その機能を実行する回路要素の組み合わせ、あるいは（h）その機能を実行するためソフトウェアを実行する適切な回路と組み合わせるいずれの形態におけるソフトウェア（従って、ファームウェア、オブジェクトコード、マイクロコード等を含む。）を含む。

【図面の簡単な説明】

【図1】本発明に従ってデータを圧縮し展開するシステム。

【図2】本発明に従ってデータを圧縮する動作の流れ図。これは図1のシステムにて有用である。

【図3】図2の動作に従って計算された入カデータストリームとフィンガプリントとを記憶するデータ構造の例。

【図4】本発明に従って符号化された一連の符号化データストリームとともに一連の非圧縮データストリームを示す。

【図5】入カデータファイルの選択部分。

【図6】本発明の原理に従って図5の入カデータファイルから符号化された符号化データファイルの選択部分。

【図7】本発明に従って展開する動作の流れ図。

【図8】本発明に従ってテキストファイルを圧縮した圧縮結果。

【図9】図2に示すように本発明に従ってデータを圧縮するC書庫ソースコードプログラム。

【図10】図9と同様。

【図11】図7に示すように本発明に従ってデータを展開するC書庫プログラム。

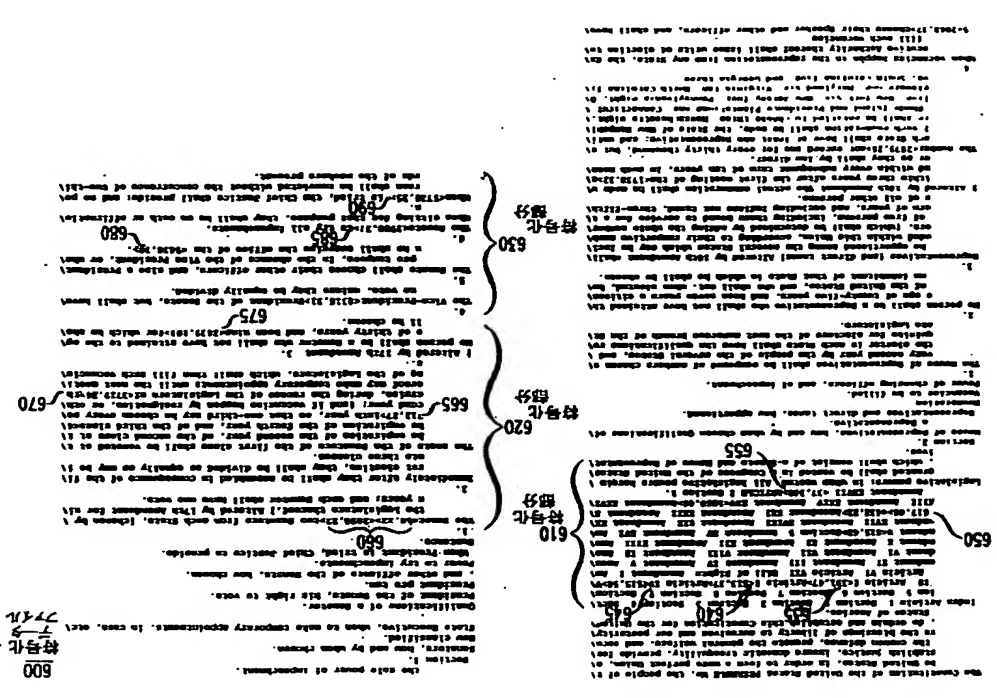
【符号の説明】

105 入カデータストリーム
110 入カデータエンコーダ

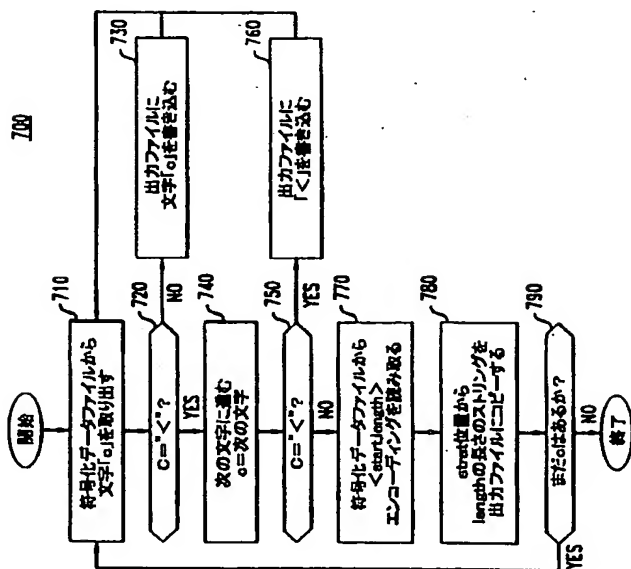
（0055）本発明の多くの原理および利点を説明するため、非常に大きなファイルに関連して本発明を適用し、そのファイルの圧縮の多くの結果を比較した。サンプルとしては、周知のCD-ROMで「Project Gutenberg Compact Disc」、Walnut Creek CDROM、Walnut Creek CAに含められた全てのテキストファイルを連結させて用いた。このCD-ROMには1994の文書が含まれている。この試験のため、我々は周知のUNIX（登録商標）の「cat」コマンドを用いて全てのテキストファイルを連結した。具体的なUNIXコマンド文字列は、「cat /s/1.xl > gull94all.txt」であった。このように連結することにより、66122kbytesの入カファイルを得て、これに本発明の原理を適用した。

（0056）図8は、連結した当該テキストファイルを圧縮した結果の比較800を示す。ファイルサイズは元圧縮した結果の比較800を示す。圧縮による元ファイルと割合をパーセントで示してあり、圧縮による元ファイルと割合をパーセントで示した。ブロックサイズb（810）を減らした効果と比較した。「com」の見出しを有する列820は、本発明を適用した効果を示してあり、入カデータのファイルのサイズの変化を示してある。「com.k」の見出しを有する列830は、ブロックサイズを変化させて調べた圧縮パーセンテージを示す。

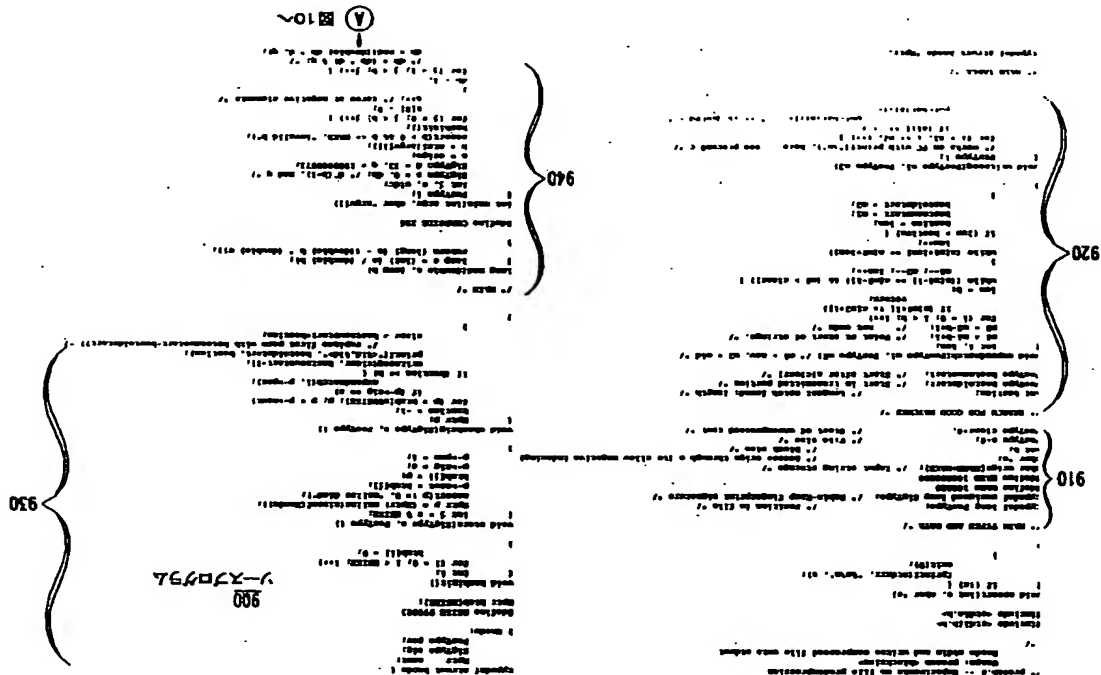
（0057）「com b lgzip.k」の見出しを有する列840および「comlgzip.k」の見出しを有する列850は、



【図7】



【図9】



[図 11]

1100 ソースコードプログラム部分

```

/* decom.o -- decompress files made by precomp.o
   Usage: precomp
   Reads stdin and writes decompressed file onto stdout
   */

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

typedef long footype;
#define MAX 10000000
char a[MAX]; /* Output string storage */
footype n=0; /* File size */

void addchar(char c)
{
    a[n++] = c;
    putchar(c);
}

int getint() /* gobble nondigit at end */
{
    int c, n = 0;
    for (;;) {
        c = getchar(); /* danger: no EOF */
        if (!isdigit(c)) break;
        n = 10*n + c - '0';
    }
    return n;
}

int main()
{
    footype i;
    int c, start, len;
    while ((c = getchar()) != EOF) {
        if (c == '<') {
            c = getchar(); /* danger: no EOF */
            if (c == '<') { /* quoted < */
                addchar('<');
            } else {
                ungetc(c, stdin);
                start = getint();
                len = getint();
                for (i = start; i < start+len; i++)
                    addchar(a[i]);
            }
        } else
            addchar(c);
    }
    return 0;
}

```

フロントページの続き

(71) 出願人 596077259

600 Mountain Avenue,
Murray Hill, New Je
rsey 07974-0636 U. S. A.

(72) 発明者 ジョン ルイス ベントリー

アメリカ合衆国、07974 ニュージャージー
ー、ニュープロビデンス、セントラル ア
ベニュー 449、アパートメント 2

(72) 発明者

マルコルム ダグラス マクローイ
アメリカ合衆国、03750 ニューハンプシ
ヤー、エトナ、ハイフィールド ロード
1